Random search global optimization using random forests

B. L. Robertson[1] C. J. Price[1] M. Reale[1]

[1]University of Canterbury, Christchurch, New Zealand

December 13, 2017

Introduction

- The bound constrained global optimization problem is of the form

$$\min f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega,$$

where the search region Ω is an n-dimensional box

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : l_j \leq x_j \leq u_j \text{ for all } j = 1, \ldots, n\}.$$

- The objective function f maps Ω into $\mathbb{R} \cup \{+\infty\}$ and is assumed to be lower semi-continuous.

- The inclusion of $\{+\infty\}$ means certain constrained problems can be considered using an extreme barrier function

$$f_\omega(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in \omega, \\ +\infty & \text{otherwise,} \end{cases}$$

where $\omega \subset \Omega$ with $m(\omega) > 0$.

(Overly-) Simplified Algorithm

1. **Initialize:** Draw a batch of points from Ω and evaluate f at each point to obtain training data $T = \{\mathbf{x}_i, f_i\}$. Set $k = 0$ and let $\mathbf{x}_0$ minimize f over T.

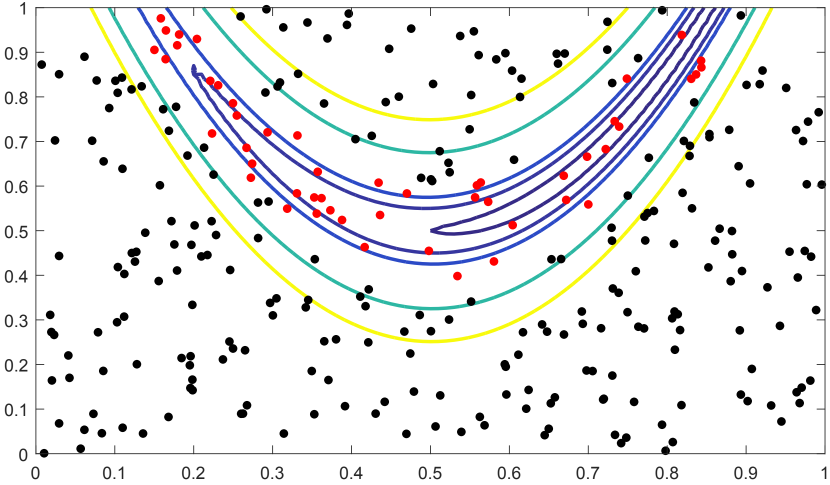
2. **Classify:** Label points in T with $f_i \leq f_t$ *low* and the remaining points *high*.

3. **Partition:** Build a random forest partition to classified T.

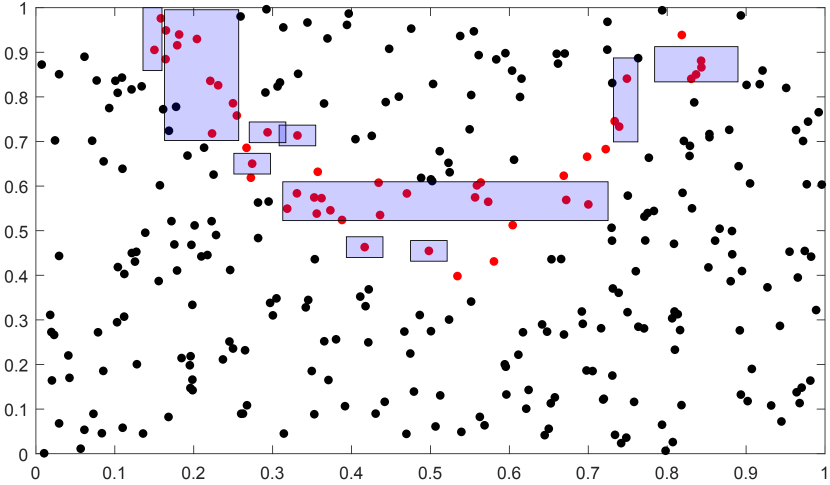
4. **Sample:** Draw a batch of points X from the low and high regions of the partition on Ω. Evaluate f at each point and let $\mathbf{x}_{k+1}$ minimize f over $T \cup X$.

5. **Update T:** Set $T \leftarrow T \cup X$, increment k and go to step 2.

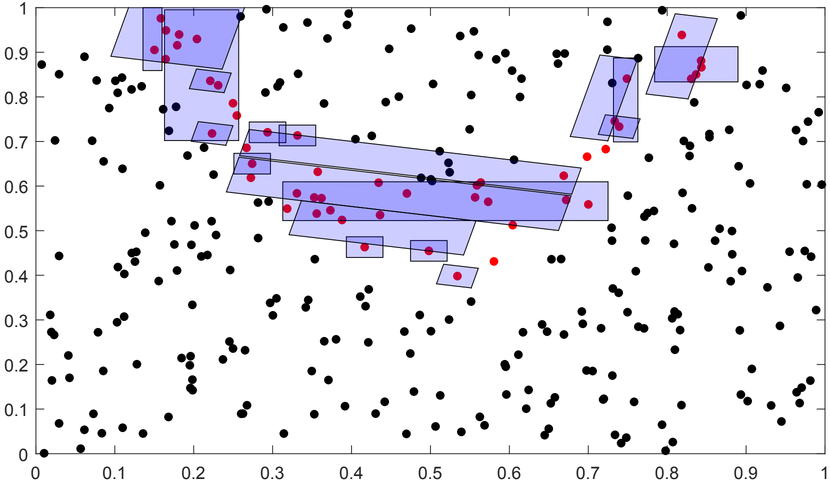
Defining training data using observed f values (red = low, black = high)



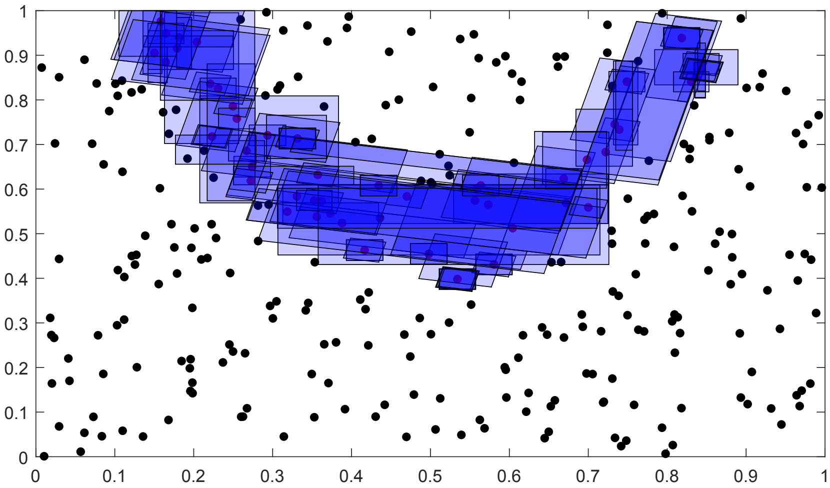
Random forest partition of Ω $(B = 1)$



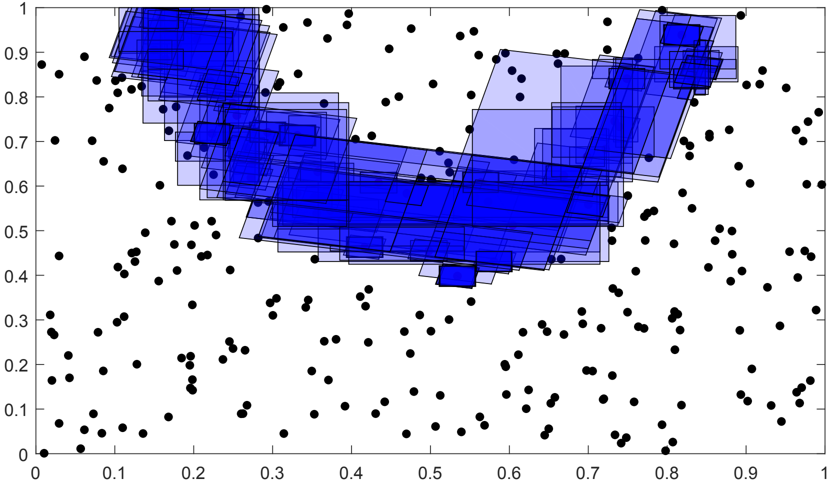
Random forest partition of Ω ($B = 2$)



Random forest partition of Ω ($B = 10$)



Random forest partition of Ω ($B = 20$)



Drawing points from the random forest partition

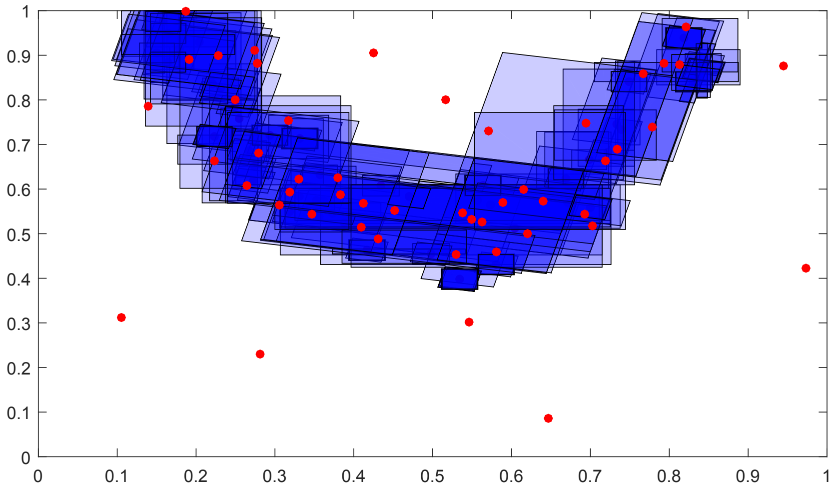
- Rather than drawing points from the high region directly, we sample Ω itself.

- To draw points from the low region, we use a three-step approach:

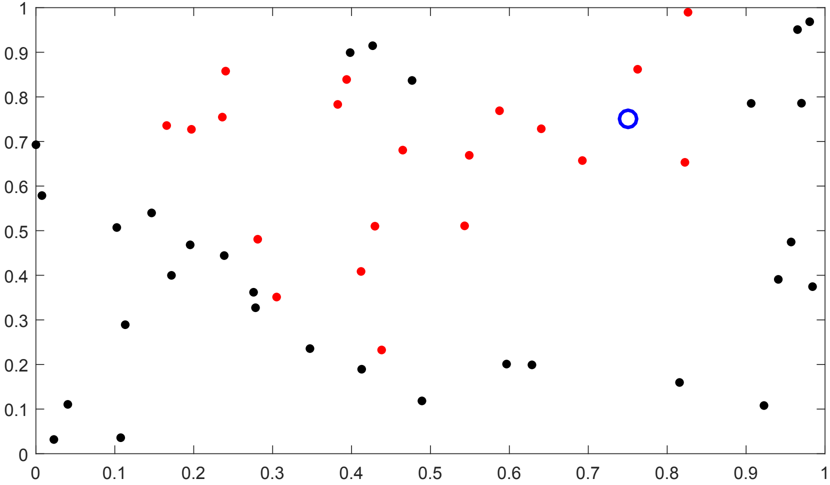
  1. Randomly choose one partition from the random forest.
  2. Randomly draw one box from the partition using selection probabilities proportional to the relative size of each box in the partition.
  3. Drawn one point from the selected box.
  4. Repeat steps 1 to 3.

- The point density will tend to be greater where the low boxes have greatest overlap, which is where the random forest is most confident that f is relatively low.

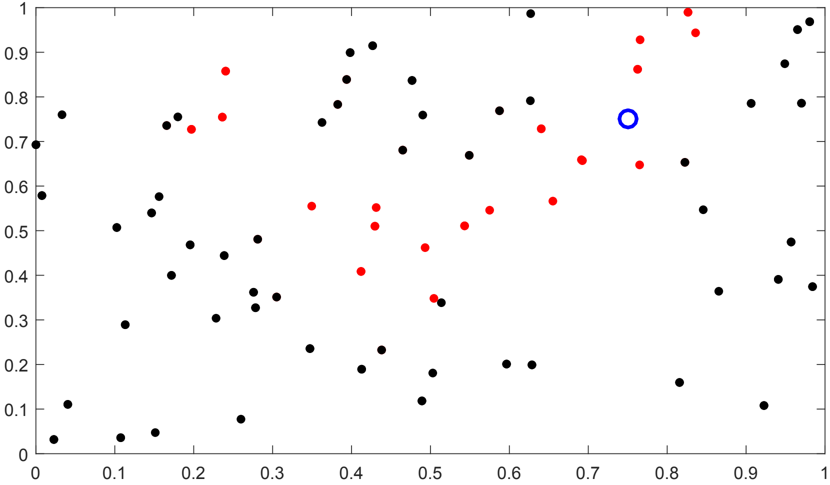
Drawing points from the random forest partition (N=50)



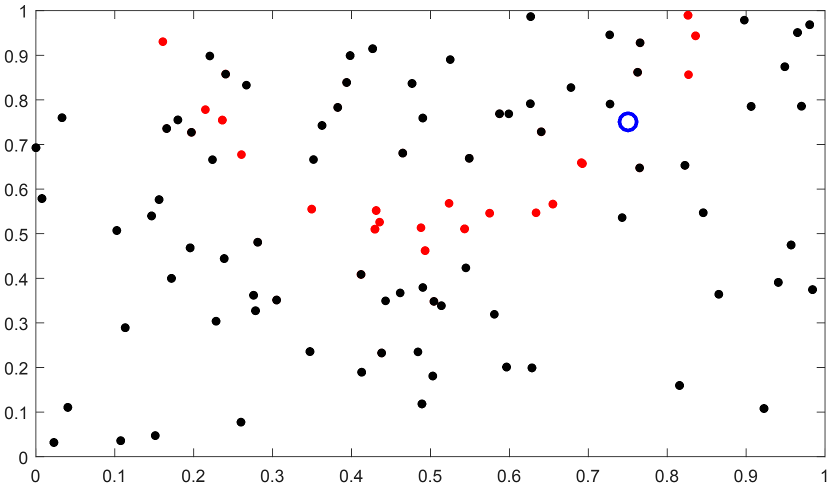
Illustrative example with $N = 25$ (Rosenbrock's function)



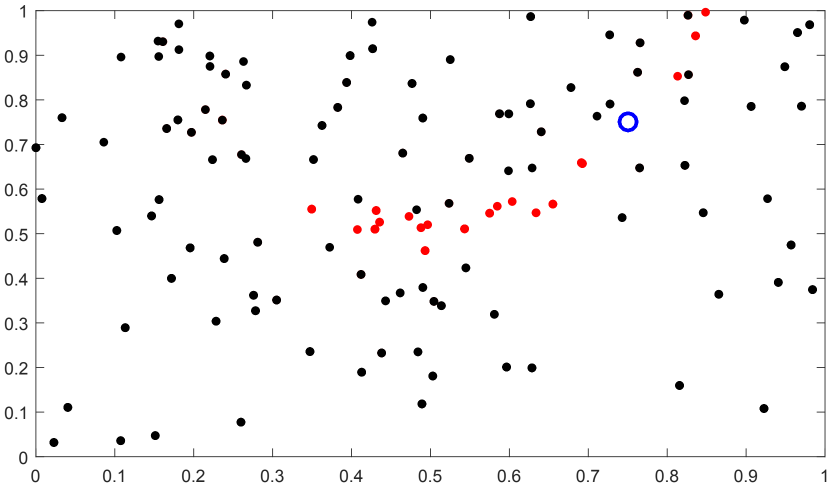
Illustrative example with $N = 25$ (Rosenbrock's function)



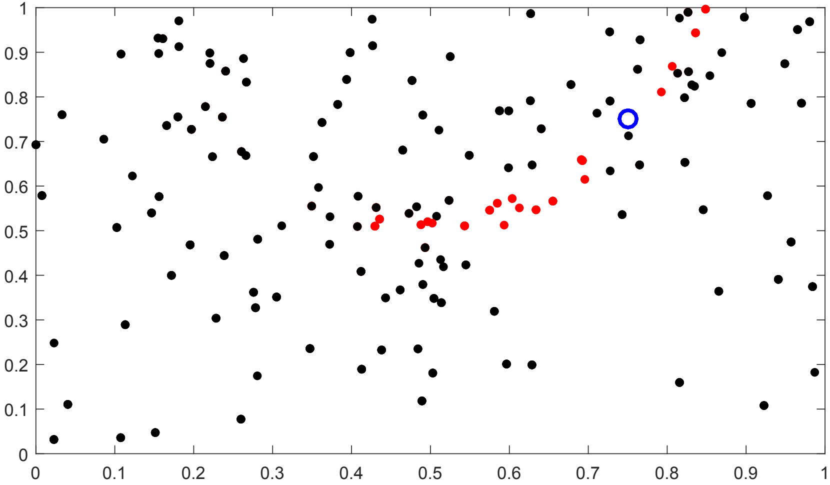
Illustrative example with $N = 25$ (Rosenbrock's function)



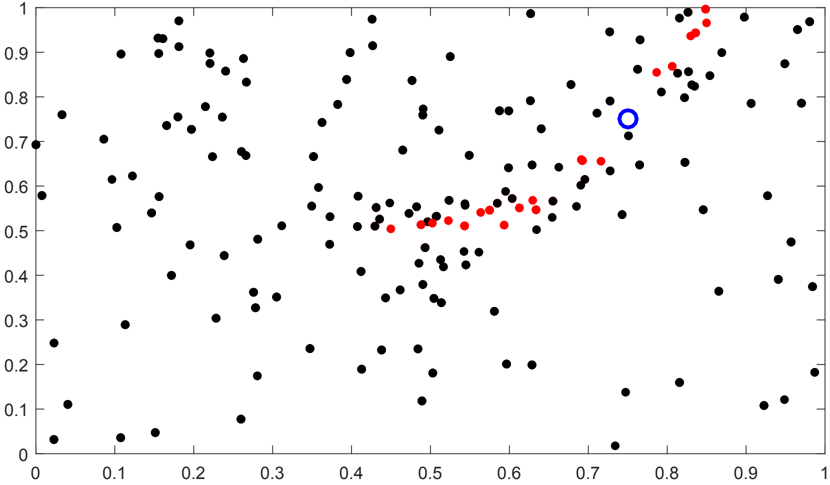
Illustrative example with $N = 25$ (Rosenbrock's function)



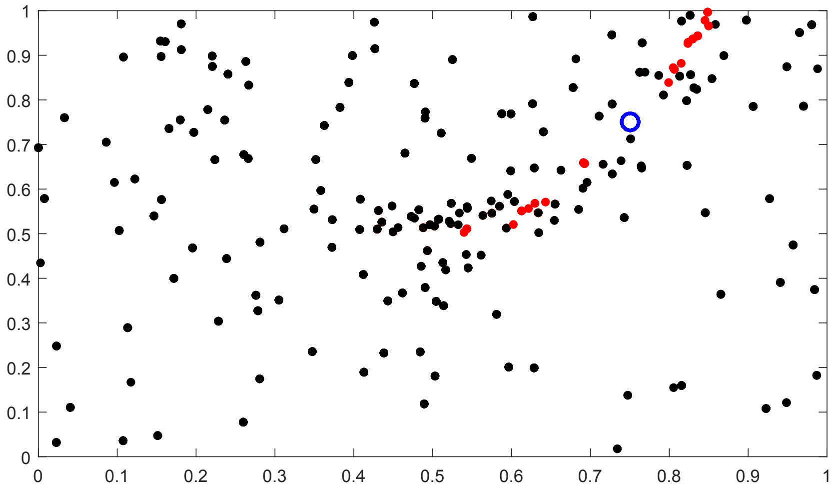
Illustrative example with $N = 25$ (Rosenbrock's function)



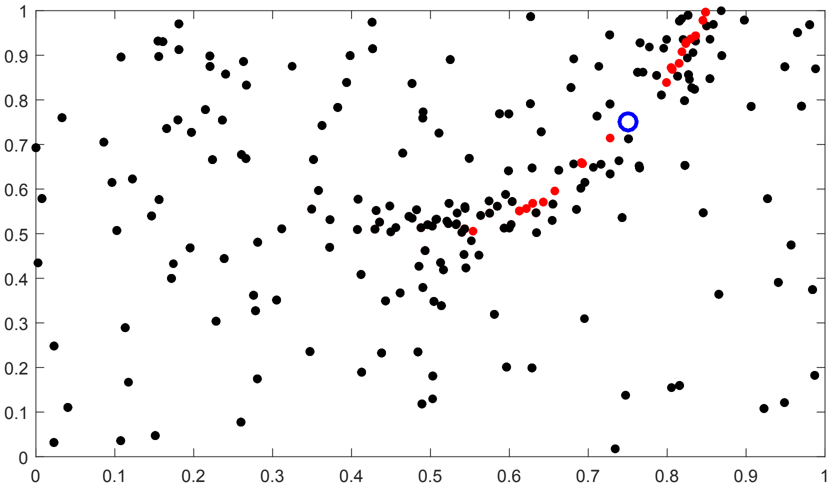
Illustrative example with $N = 25$ (Rosenbrock's function)



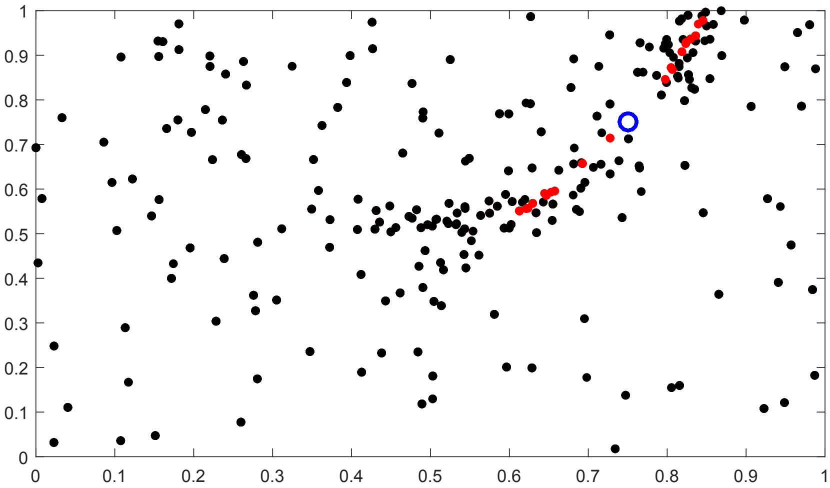
Illustrative example with $N = 25$ (Rosenbrock's function)



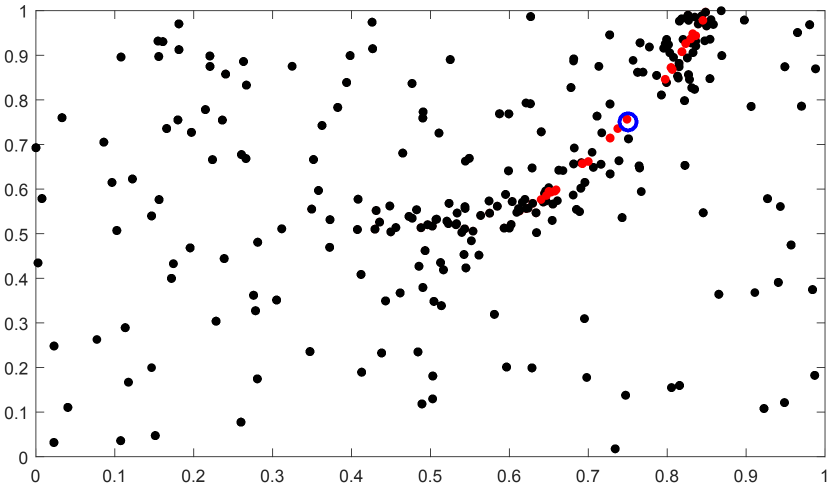
Illustrative example with $N = 25$ (Rosenbrock's function)



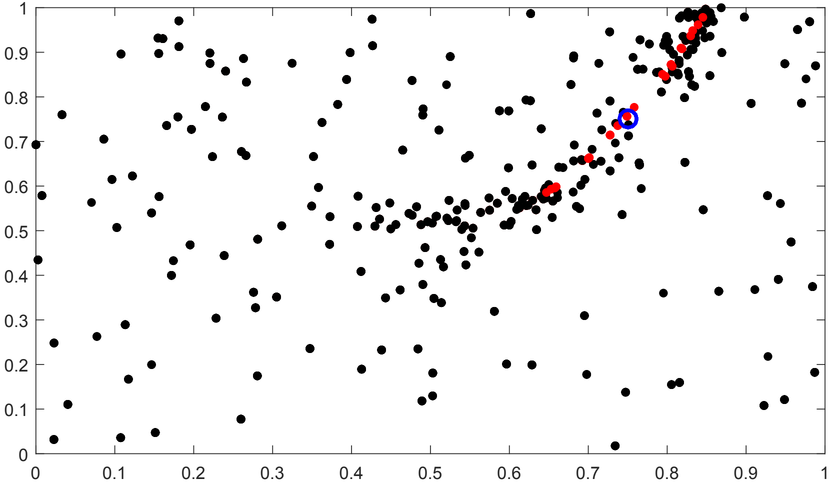
Illustrative example with $N = 25$ (Rosenbrock's function)



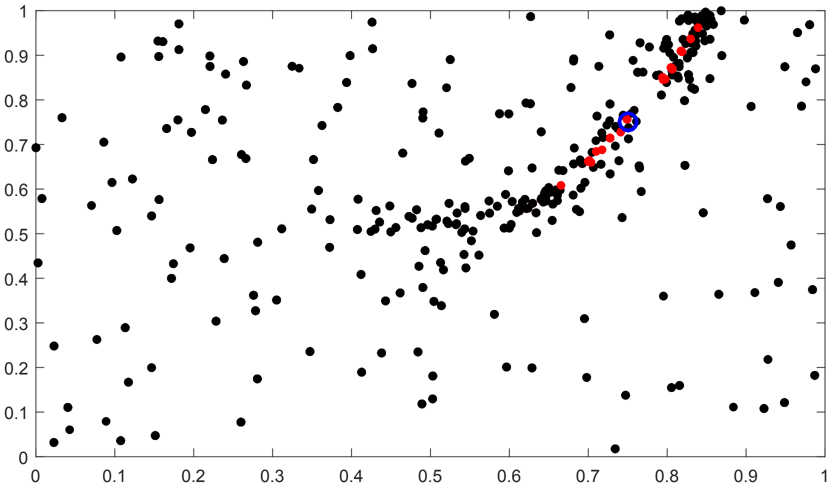
Illustrative example with $N = 25$ (Rosenbrock's function)



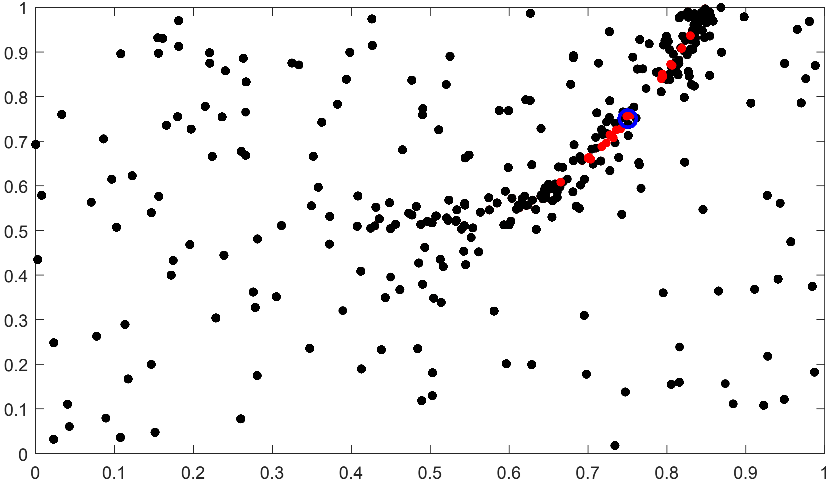
Illustrative example with $N = 25$ (Rosenbrock's function)



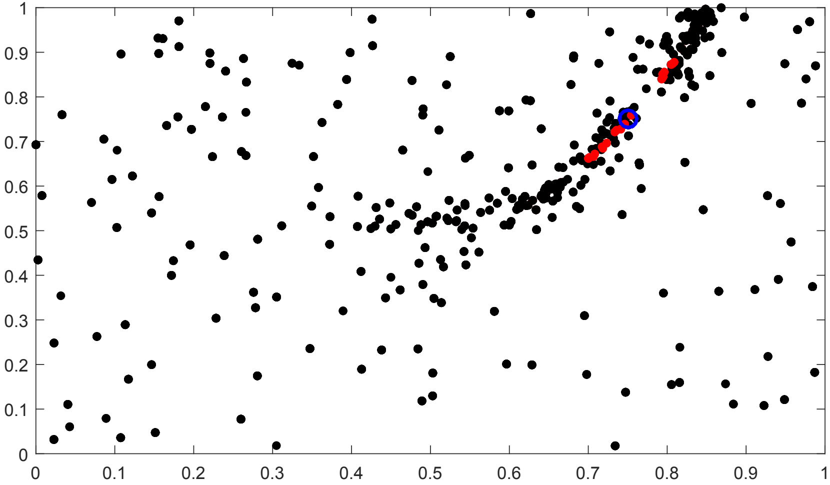
Illustrative example with $N = 25$ (Rosenbrock's function)



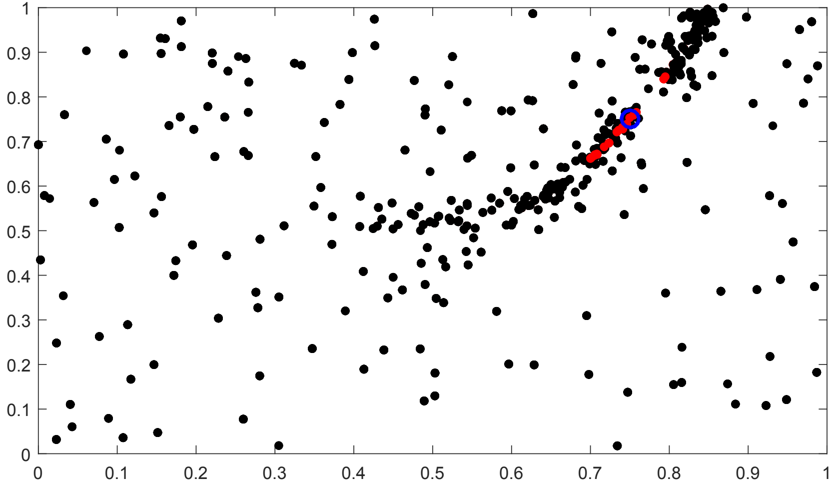
Illustrative example with $N = 25$ (Rosenbrock's function)



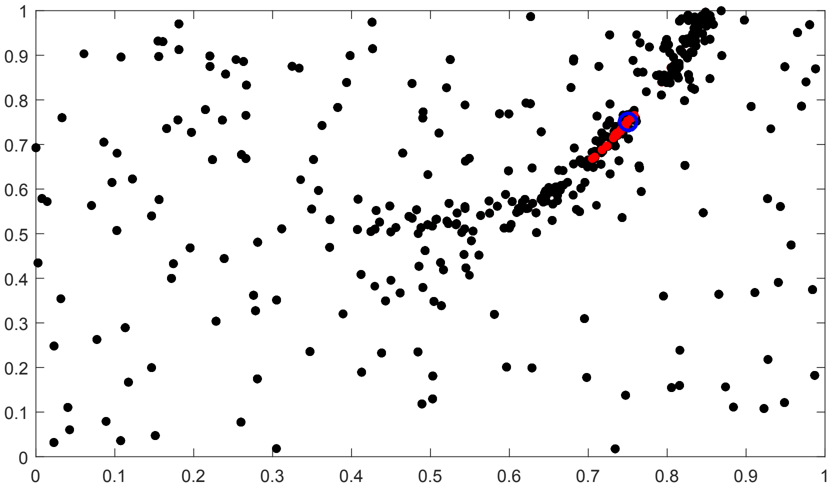
Illustrative example with $N = 25$ (Rosenbrock's function)



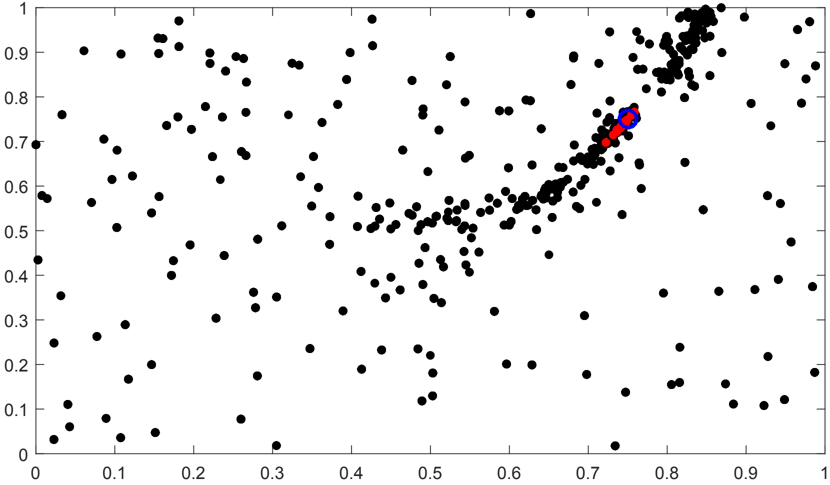
Illustrative example with $N = 25$ (Rosenbrock's function)



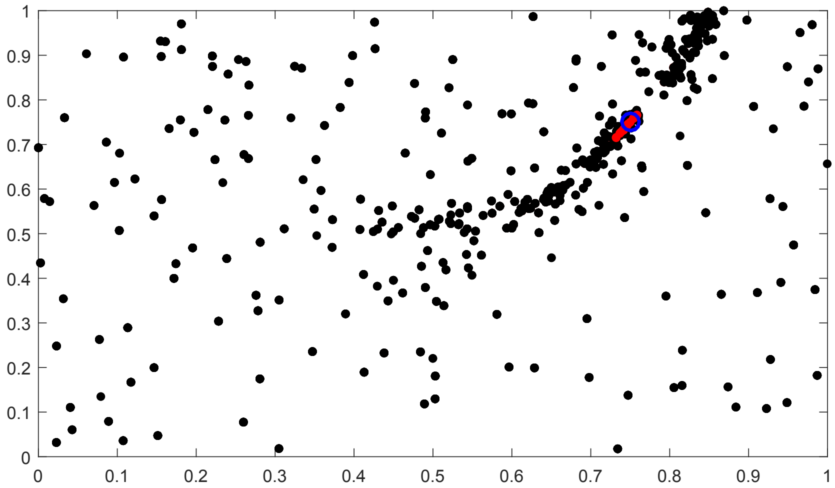
Illustrative example with $N = 25$ (Rosenbrock's function)



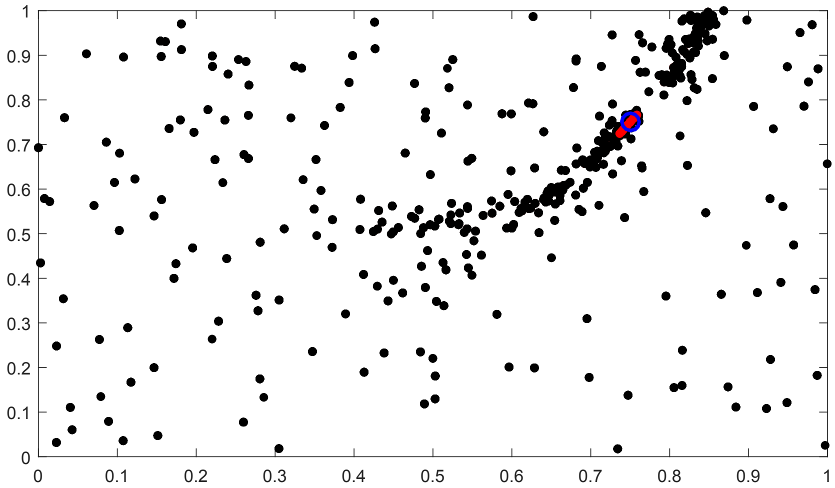
Illustrative example with $N = 25$ (Rosenbrock's function)



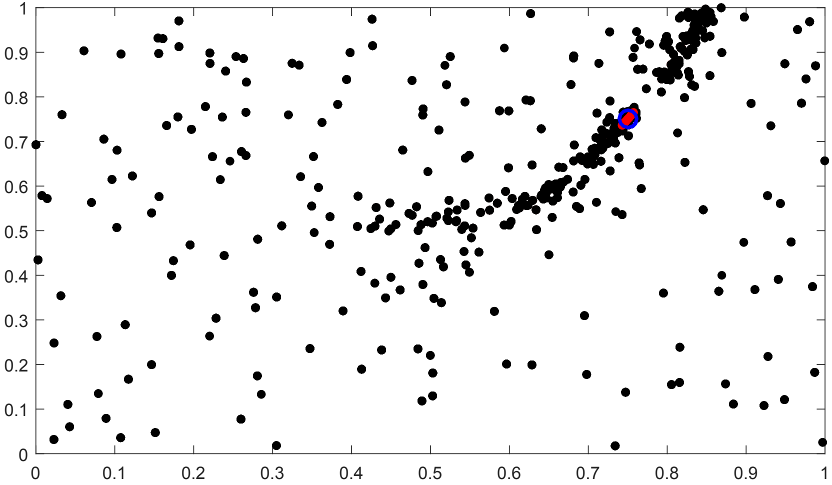
Illustrative example with $N = 25$ (Rosenbrock's function)



Illustrative example with $N = 25$ (Rosenbrock's function)



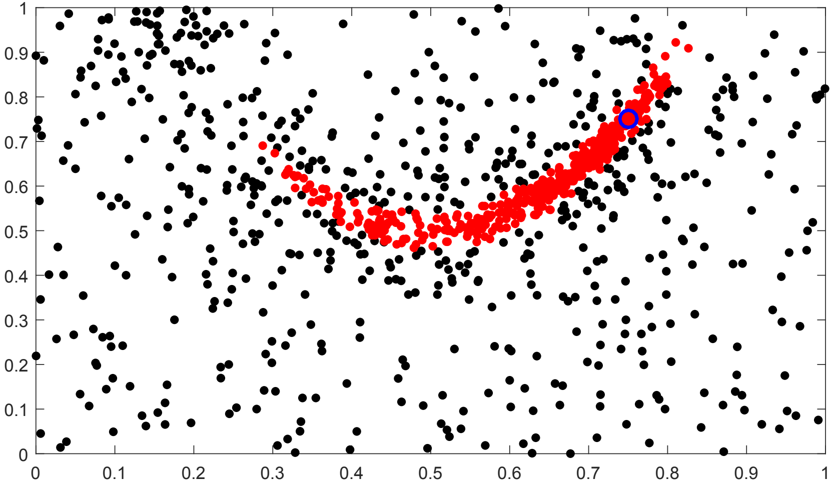
Illustrative example with $N = 25$ (Rosenbrock's function)



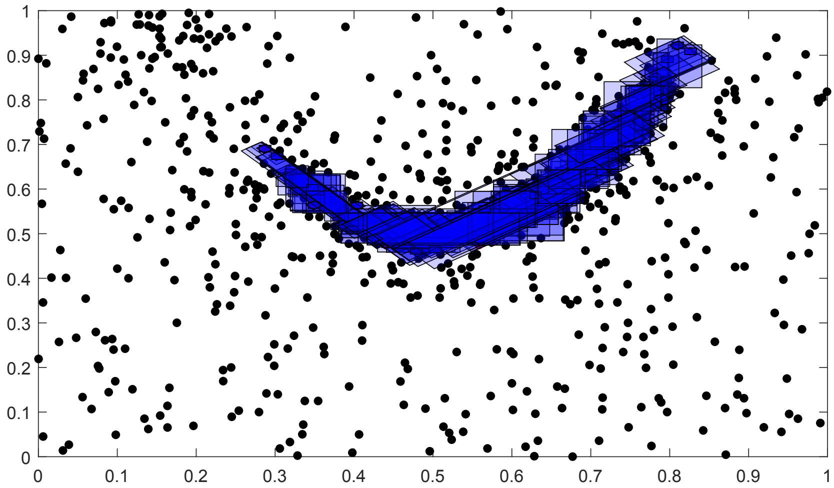
Restarts

- There is a balancing act between the rate of convergence to a local minimum and missing the global minimum.
- Restarting the algorithm from time to time reduces the risk of missing the global minimum.
- Two simple approaches are:
  1. Restart each time a better point is found (or if sufficient descent is made).
  2. Restart when a minimum low region size is achieved (or sequence of sizes).

Recycling points after a restart



Recycling points after a restart



Convergence

- **Definition:** *A point* $\mathbf{x}_* \in \Omega$ *for which the set*

$$L(\mathbf{x}_*) = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) < f(\mathbf{x}_*)\}$$

  has Lebesgue measure zero is called an essential global minimizer of f*.*

- If f is lower semi-continuous and bounded below, the sequence of best points generated by the algorithm converges to an essential global minimizer of f with probability one.

Conclusion

- A global optimization algorithm that alternates between partition and sampling phases has been presented.
- At each partition phase a random forest is used to predict where f is likely to be low. Points are evaluated in these regions to direct the search in promising regions.
- The method is provably convergent (under mild conditions) on smooth and non-smooth problems.
- Although not presented here, our method is competitive on a number of different smooth and non-smooth test problems ranging in dimension from 2 to 10.